

**In the Claims:**

1. (currently amended) A remote data processing system comprising:  
a data receiver for receiving a data message;  
a remote software module arranged to receive the data message from the data receiver, the remote software module including at least a first stage software component cascaded with a second stage software component; and  
a fault detector associated with the first software stage component and the second software stage component to detect a fault in the remote software module by detecting whether the data message or a derivative thereof flows through at least one of the first stage software component and the second stage software component.
2. (original) The system according to claim 1 further comprising a remote status reporter for reporting a status message on at least one of the remote software module and hardware of the remote data processing system.
3. (original) The system according to claim 1 further comprising a database for storing detected faults, stage identifiers, and fault descriptions outputted by the fault detector.
4. (original) The system according to claim 1 further comprising a database for storing status data on corresponding components of a remote data processing system.
5. (original) The system according to claim 1 further comprising a database, the fault detector logging one or more error messages into the database.

6. (original) The system of claim 1 further comprising a feedback generator associated with the remote software module, the feedback generator receiving a status code outputted from the remote software module and forwarding the status code to a transmitter for transmission via a communications network.

7. (original) The system according to claim 1 further comprising a feedback generator associated with the remote software module, the feedback generator generating a status code for a transmitter upon detection of a status code from the remote software module.

8. (original) The system according to claim 1 wherein the fault detector has logical connections including a connection with an input of the first software stage component, an output of the first software stage component, and an output of second software stage component.

9. (currently amended) The system according to claim 8 wherein the fault detector identifies the first software stage as a faulty software component if the data message is present at an input of the first software stage, but not the output of the first software stage.

10. (currently amended) The system according to claim 8 wherein the fault detector identifies the second software stage as a faulty software component if the data message is present at an input of the second software stage, but not the output of the second software stage.

11. (currently amended) The system according to claim 8 wherein the fault detector identifies the first software stage as a faulty software component if a derivative of the data message is present at an input of the first software stage, but not the output of the first software stage.

12. (currently amended) The system according to claim 8 wherein the fault detector identifies the second software stage as a faulty software component if a derivative of the data message is present at an input of the second software stage, but not the output of the second software stage.

13. (currently amended) A method for monitoring a remote data processing system, the method comprising:

having a remote data processing system receive ~~receiving~~ a data message from a base data processing system via a communications network that is external to the remote data processing system;

cascading at least a first stage software component and a second stage software component to form an installed remote software module of the remote data processing system for accepting the received data message;

detecting the data message or a derivative at a group of logical nodes within the installed remote software module to determine flow of the data message, or a derivative thereof, between the logical nodes and, hence, flow through at least one of the first stage software component and the second stage software component; and

identifying a deficient software component of the installed remote software

module as any of said software stage components that blocks or disrupts the flow of the data message between two adjacent logical nodes.

14. (currently amended) The method of claim 13 further comprising:

passing a status code, distinct from the data status message, from at least an input of the installed remote software module to an output of the installed remote software module to indicate that the installed remote software module is operational.

15. (currently amended) The method of claim 13 further comprising:

passing a status code from at least an input of the [a] communications network to an output of the communications network to indicate that the communications network is operational.

16. (currently amended) The method of claim 13 further comprising:

routing the status code from the base data processing system via the communications network to the [a] remote data processing system ~~associated with the installed remote software module~~; and

routing the status code from the remote data processing system to the base data processing system via the communications network to indicate that the continuity of at least one logic data path traversed by the status code.

17. (currently amended) The method according to claim 13 further comprising:

tapping into a logical data path between the first stage software component and

the second stage software component stages to detect whether each of the first and second stage software components ~~component stages~~ are functioning.

18. (currently amended) The method according to claim 13 further comprising:  
determining that one of the at least a first stage software component and a second stage software component immediately following the last detected data message is at fault.

19. (currently amended) The method according to claim 13 further comprising:  
assigning stage identifiers to distinguish the at least a first stage software component and a second stage software component ~~one stage~~ from one another and to identify a faulty stage.

20. (currently amended) The method according to claim ~~19~~ 13 further comprising:  
associating a fault description with each of the stage identifiers ~~identifier~~ for transmission to a management system via a communications network.

21. (original) The method according to claim 13 further comprising:  
archiving a fault analysis report in a database associated with the remote processing system.

22. (currently amended) A method of monitoring a business-to-business system, the method comprising:

transmitting a status code from a base data processing system to a remote data processing system via a communications network;

receiving the status code at a data receiver in the remote data processing system;

inputting the status code into a remote software module of the remote data processing system;

determining whether the remote software module provides a logical data path of continuity to the status code;

outputting the status code from an output of the remote software module if the determining determines that the remote software module provides a logical data path of continuity to the status code; and

transmitting the outputted status code back to the base data processing system via the communications network as feedback indicative of the proper end-to-end continuity of communications in a business-to-business environment.

23. (original) The method according to claim 22, the method further comprising:

storing the status code from an output of the remote software module as a dummy transaction in the database; and

retrieving the status code as the dummy transaction in the database and feeding the retrieved status code for transmission to the base data processing system if the database provides a logical data path of continuity for the status code.